Course code	Course Title	L	T	P	J	С
CSI2007	SOFTWARE ENGINEERING PRINCIPLES	2	0	2	0	3
Pre-requisite	Nil	Syllabus version v.1.0		1		

Course Objectives:

- 1.To introduce the essential software engineering concepts involved in developing software products and components
- 2. To impart development skills during design, implementation and testing of reliable software systems across various disciplines
- 3. To familiarize engineering practices and standards used in developing software products and components

Expected Course Outcome:

- 1. Apply the principles of Software engineering methodology during software development and deployment process.
- 2. Document various processes like Requirement Engineering, Design and Testing.
- 3. Demonstrate an ability to use the techniques and tools necessary for significant application domains
- 4. Apply software testing and quality knowledge and engineering methods for various applications
- 5. Analyze the effectiveness of managing software projects through various techniques like Estimations, Scheduling and Quality Models
- 6. Apply benchmarking standards in process and in product.

Module:1	INTRODUCTION	5 hours

Software Engineering- Need, Importance and its characteristics - Software Process- Generic process model-Prescriptive process model-specialized, unified process-Agile development-Agile Process- Extreme Programming- Other agile Process models-Software engineering Knowledge-core Principles-Principles that guide each framework Activity.

Module:2 | SOFTWARE REQUIREMENT ANALYSIS

5 hours

Requirements Engineering-Establishing the Groundwork-Eliciting Requirements- Developing use cases-Building the requirements model-Negotiating, validating Requirements-Requirements Analysis-Requirements Modeling Strategies.

Specifying Requirements: functional and non-functional requirements; specification exercise. Managing the Requirements Process: methods which provide a structure for co-operation between different stake holders. Prototyping: The role of prototyping in requirements techniques for prototyping. Requirements for Future Technologies: Computer Supported Co-operative Work (CSCW); networked multi-media systems.

Module:3 | **SOFTWARE DESIGN**

5 hours

Design concepts and principles - Abstraction - Refinement - Modularity - Cohesion & coupling, Architectural design, Detailed Design - Transaction & Transformation, Refactoring of designs, Object-oriented Design User-Interface Design; Object Oriented Design Concepts and Diagrams - Use Case Diagrams - Class Diagrams - Interaction Diagrams - State chart Diagrams - Activity Diagrams - Package Diagrams - Component Diagrams - Deployment Diagrams

Module:4 | SOFTWARE IMPLEMENTATION

4 hours

Structured coding Techniques-Coding Styles-Standards and Guidelines- Documentation Guidelines-Modern Programming Language Features: Type checking-User defined data types-Data Abstraction-Exception Handling- Concurrency Mechanism – Seven Steps of implementing software – Implementation Challenges and its resolution.

Module:5 | **SOFTWARE TESTING**

4 hours

TESTING: Introduction; Software Testing Fundamental; Testing Principles; Testing Levels; Verification and Validation: Validation Testing, Validation Test Criteria; Test Plan: Test Documentation; Test Strategies: Top-Down Testing, Bottom-Up Testing, Thread testing, Stress testing, Back-to-back testing; Testing methods and tools: Testing through reviews, Black-box testing (Functional testing), White box testing (glass-box testing), Testing software changes; Additional requirements in testing OO Systems; Metrics Collection, Computation, and Evaluation; Test and QA plan; Managing Testing Functions.

Module:6 | SOFTWARE MAINTENANCE

3 hours

Software Maintenance, Types of Maintenance, Structured versus unstructured maintenance – Maintenance costs – Typical problems with maintenance and its side-effects – Maintenance

process - Software Configuration Management – Component Reusability - Overview of REengineering & Reverse Engineering- Business Process Reengineering- Restructuring- Forward Engineering- Economics of Reengineering.

Module:7	PROJECT PLANNING AND RISK	2 hours
	MANAGEMENT	

Objectives of Activity planning – Project schedules – Activities – Sequencing and scheduling – Network Planning models – Forward Pass & Backward Pass techniques – Critical path (CRM) method – Risk identification – Assessment – Monitoring – PERT technique – Monte Carlo simulation – Resource Allocation – Creation of critical patterns – Cost schedules.

Modu	le:8 RECENT TRENDS	2 hours
	Total Hours	30 Hrs
Lab E	xperiments	
	ork Break-down Structure (Process Based, Product Based, Geographic sed and Role Based)	30 Hrs
2. Es	imations – Cost & Schedule	
3. En	tity Relationship Diagram, Context flow diagram, DFD (Structural	
Mo	deling and Functional Modeling)	
4. Sta	te Transition Diagrams (Behavioral Modeling)	
5. Sy	stem Requirements Specification	
6. UN	IL diagrams for OO Design	
7. To	ols for Version Control	

Text Book(s)

1. Roger Pressman and Bruce Maxim, Software Engineering: A Practitioner's Approach, 9th Edition, McGraw-Hill, 2020.

Reference Books

1. Ian Sommerville, Software Engineering, 10 th Edition, Addision-Wesley, 2015

8. Black-box, White-box testing Non-functional testing

- 2. Pankaj Jalote, An Integrated Approach to Software Engineering (Texts in Computer Science), Reprint Springer, 2010
- 3. William E. Lewis, "Software Testing and Continuous Quality Improvement", Third Edition, Auerbach Publications, 2008
- 4. David Gustafson, Schaum's Outline of Software Engineering,1st Edition, 2020

Mode of Evaluation: CAT / Assignment / Quiz / FAT / Project / Seminar/Lab

Recommended by Board of Studies	11-02-2021		
Approved by Academic Council	No. 61	Date	18-02-2021